Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 3, June 2026, pp. 2758~2771
ISSN: 2302-9285, DOI: 10.11591/eei.v15i3.11889

Detecting anomalies in MQTT/MQTT-SN traffic using intelligent learning models

Nabeel Mustafa Alassaf[1], Selvakumar Manickam[1], Ammar Odeh[2], Mohammed Anbar[1]
[1]Cybersecurity Research Centre, Universiti Sains Malaysia (USM), George Town, Malaysia
[2]Department of Computer Science, King Hussein School of Computing Sciences, Princess Sumaya University for Technology (PSUT), Amman, Jordan

Article Info

Article history:

Received Nov 19, 2025
Revised Apr 1, 2026
Accepted Apr 19, 2026

Keywords:

Anomaly detection
Edge computing
eXtreme gradient boosting
Internet of things security
Machine learning
Message queuing telemetry transport
Message queuing telemetry transport-sensor network

ABSTRACT

The widespread adoption of the internet of things (IoT) has heightened demand for secure, efficient communication across constrained devices. Lightweight protocols such as message queuing telemetry transport (MQTT) and its variant MQTT-sensor networks (SN) are widely used for IoT messaging but lack intrinsic security mechanisms, leaving them vulnerable to denial-of-service, spoofing, and injection attacks. This study presents a machine learning (ML)-based anomaly detection framework designed to enhance the security of MQTT and MQTT-SN traffic. We emulate realistic IoT environments to generate both benign and malicious traffic, extracting protocol-specific features such as packet length, topic length, quality of service (QoS) levels, and publish frequency. Three supervised models—random forest (RF), eXtreme gradient boosting (XGBoost), and long short-term memory (LSTM)—were trained and evaluated using cross-validation and statistical performance metrics. Experimental findings demonstrate that XGBoost achieved the best overall results, with 97.4% accuracy, 95.9% F1-score, and low false-positive and false-negative rates. Furthermore, the framework was successfully deployed on edge devices such as Raspberry Pi Zero W and ESP32, confirming its real-time feasibility and efficiency. The proposed approach highlights the potential of intelligent learning models to deliver lightweight, deployable, and effective intrusion detection for IoT systems utilizing MQTT and MQTT-SN communication protocols.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Selvakumar Manickam
Cybersecurity Research Centre, Universiti Sains Malaysia (USM)
George Town, Penang, Malaysia
Email: selva@usm.my

1. INTRODUCTION

The internet of things (IoT) continues to grow rapidly, connecting billions of devices across sectors such as healthcare, smart homes, industrial automation, and transportation [1], [2]. To fulfill the communication requirements of these resource-constrained environments, lightweight messaging protocols, including the message queuing telemetry transport (MQTT) and its sensor network variant, MQTT-sensor networks (SN), have been widely adopted [3]. These protocols have minimal bandwidth consumption and use little power, making them appropriate for constrained devices. However, due to their simplicity and lack of security features, they are vulnerable to various attacks, including denial-of-service (DoS) attacks, message spoofing, and unauthorized access [4], [5].

Conventional safety mechanisms, such as encryption and authentication, are often not well-suited for IoT applications due to constraints on processing and energy [6]. As a consequence, machine learning (ML)

solutions have emerged as a viable alternative to anomaly detection and intrusion prevention in IoT communication systems [7], [8]. By learning the behavioral patterns of network traffic, intelligent models can detect subtle deviations from expected behavior, even when the attack behavior is unknown [9].

This paper presents a comprehensive ML-based approach to identifying anomalies in MQTT and MQTT-SN traffic that remains efficient, accurate, and deployable in real-world scenarios. Various intelligent learning models (eXtreme gradient boosting (XGBoost), random forest (RF), and long short-term memory (LSTM) neural network (NN)) are evaluated to determine which solutions are best suited for these lightweight protocols [10], [11].

While several studies have proposed ML-based intrusion detection systems (IDS) for IoT traffic, most focus on general network-level features and do not address the protocol-specific characteristics of MQTT or MQTT-SN. Moreover, existing works often overlook the challenges of deploying such models on resource-constrained edge devices. In contrast, our work contributes a protocol-aware anomaly detection framework that not only extracts fine-grained features from MQTT/MQTT-SN traffic but also evaluates the feasibility of deploying these models on real-world IoT hardware [12], [13]. Additionally, we introduce a temporal detection approach using LSTM to model publish/subscribe sequence patterns—an aspect rarely explored in existing MQTT-based IDS frameworks.

Despite the growing number of intrusion/anomaly detection studies targeting IoT communications, the research gap remains insufficiently addressed for lightweight publish–subscribe environments such as MQTT and MQTT-SN. Most existing works primarily rely on generic network/flow features and report detection accuracy in controlled settings, while providing limited evidence on protocol-aware feature engineering that captures the unique control fields, session behaviors, and message-level patterns of MQTT/MQTT-SN. In addition, prior studies often evaluate a single protocol (typically MQTT) and rarely validate whether the proposed models remain effective and practical when deployed on resource-constrained edge devices, where latency, CPU/RAM usage, and energy overhead are critical. Therefore, there is a clear need for a comparative, protocol-specific detection framework that simultaneously: i) models MQTT and MQTT-SN behaviors using protocol-level features and ii) demonstrates feasibility through deployment-oriented performance evaluation, thereby bridging the gap between high reported accuracy and real-world IoT operational constraints.

This work presents the following key contributions:

- Protocol-specific feature engineering: we extract and analyze unique features from MQTT and MQTT-SN traffic-such as topic length, message type, publish/subscribe frequency, and quality of service (QoS) level - to improve detection accuracy in a protocol-aware manner.
- Comparative evaluation of ML models: we implement and evaluate various supervised ML algorithms, including XGBoost, RF, and LSTM, to detect anomalies in real and emulated MQTT/MQTT-SN environments. Our analysis highlights trade-offs between accuracy, computational cost, and suitability for deployment on edge devices.
- Lightweight and deployable intrusion detection framework: we design an anomaly detection system that balances security performance with resource efficiency, enabling deployment on low-power IoT devices such as the ESP32 and Raspberry Pi Zero W. Our framework achieves high detection rates with minimal impact on system resources.

2. MATERIALS

To situate the proposed framework within the broader research landscape, this section reviews key studies addressing lightweight encryption, intrusion detection, and secure communication in IoT and MQTT/MQTT-SN environments. The discussion highlights recent advances, identifies existing limitations, and establishes the motivation for developing a more efficient and adaptive security solution tailored to resource-constrained IoT devices [14].

The algorithm proposed by Has *et al.* [15], which integrates Huffman coding and AES encryption, presents opportunities for further refinement and broader application. Future research could explore adaptive compression techniques, such as dynamic Huffman coding or hybrid approaches, to optimize performance across diverse agricultural IoT datasets. Enhancing security through lightweight cryptographic alternatives while maintaining robustness could further reduce computational demands on resource-constrained devices. Additionally, incorporating edge computing for localized data preprocessing or blockchain for immutable audit trails could improve scalability and trust in distributed agricultural networks. Evaluating energy efficiency trade-offs across hardware platforms and standardizing the approach for interoperability with emerging IoT protocols would strengthen its practical adoption. These extensions could position the algorithm as a versatile solution not only for precision agriculture but also for other IoT domains requiring efficient and secure data management.

Palmese *et al.* [16] propose a lightweight anomaly-based IDS for MQTT sensor networks that trains an radial basis function (RBF)-kernel support vector machine (SVM) on an MQTT-specific dataset targeting DoS, wildcard-subscription misuse, and MitM attacks; their pipeline reduces features, standardizes inputs, and tunes hyperparameters to push accuracy toward ~99%, while showing that aggressive feature pruning can erode intrusion-class detection and that the resulting model remains feasible for deployment on constrained devices.

Roldán-Gómez *et al.* [17] present an exhaustive security assessment of MQTT-SN, noting that features such as numeric TopicId, sleeping clients, and UDP transport (which eases IP spoofing) expand the attack surface beyond MQTT. The authors implement and test three concrete attacks: a Disconnect Wave that repeatedly re-CONNECTs with hijacked client IDs to eject legitimate nodes, a Spoofing-via-ID attack that sets clean session=1 to erase a device's subscriptions and re-subscribe it to attacker-chosen topics, and a Wake-Up Wave that spams PINGREQ to keep sleeping nodes awake and drain resources.

Mofidul *et al.* [18] propose a secure, integrated industrial IoT (IIoT) infrastructure that combines edge and cloud computing with AI for real-time acquisition of energy data, anomaly detection, and monitoring. Their system features smart data acquisition devices (SDADs) for collecting heterogeneous sensor data, secured communication via TLS and MQTT protocols, and PostgreSQL databases for handling big data. A major innovation was the use of individual-isolation forest AI models to detect anomalies across different datasets, achieving an average accuracy of 92%. The system also included real-time dashboards for monitoring and open-source tools such as Mosquitto and node-RED to minimize costs. Arguably, the most notable outcome is the development of a globally accessible, AI-enhanced IIoT framework that is both secure (protecting data) and efficient (processing data without burdening the internet and providing real-time insights), and aligned with the smart energy management and Industry 4.0 frameworks.

Gupta *et al.* [19] present an end-to-end encryption mechanism for securing MQTT communications in Industry 4.0 (I4.0) using ciphertext-policy attribute-based encryption (CP-ABE). They introduce MQTTS, a framework that uses the fast attribute message encryption (FAME) CP-ABE scheme to protect the confidentiality and integrity of data while remaining compatible with the MQTT standard. The system features a transparent encryption layer that operates without modifying the MQTT header, enabling seamless interoperability with existing deployments. Key innovations include dynamic access policies for secure data sharing and efficient key management using elliptic curve cryptography. Performance evaluations on resource-constrained devices, such as the Raspberry Pi, demonstrate the scheme's feasibility, with encryption times and ciphertext sizes analyzed across different policy configurations. The most significant achievement is the development of a lightweight, scalable, and secure MQTT wrapper that addresses the lack of end-to-end encryption in IIoT ecosystems, making it suitable for dynamic I4.0 environments. Future work may explore key revocation and distributed CP-ABE schemes to optimize them further.

Hwang *et al.* [20] analyzed MQTT performance using Mosquitto as a broker, demonstrating satisfactory performance across various load conditions and highlighting the importance of defining security protocols. Siddharthan *et al.* [21] proposed an intrusion detection system using ensemble ML on the SENMQTTSET dataset, achieving over 99% accuracy. Alaiz-Moreton *et al.* [22] applied deep learning and ensemble methods to MQTT-based IoT attack detection, achieving 96.08% accuracy with gated recurrent units (GRUs), while Yalli *et al.* [2] compared multiple ML algorithms for IoT attack detection, with RF achieving 99.4% accuracy.

Recent studies have continued to explore lightweight and intelligent security mechanisms tailored for constrained IoT environments. Cai et *al.* [23] introduced a hybrid convolutional neural network (CNN)-GRU model for anomaly detection in industrial control systems, achieving high detection accuracy but requiring significant computational resources, limiting edge deployment. Similarly, Vansiya *et al.* [24] proposed a federated learning-based IDS that preserves privacy across distributed smart home devices but involves frequent synchronization and communication overhead. Almufareh *et al.* [25] developed an edge-optimized AutoEncoder for anomaly detection in long-range wide area network (LoRaWAN)-based IoT networks, which demonstrated fast convergence and low memory usage. Other works, such as Alhanif and Ilyas [26], have evaluated real-world MQTT attack datasets to benchmark IDS performance, suggesting that topic-specific features and payload entropy can improve detection rates for spoofing attacks. Additionally, Alketbi and Mehmood [27] presented an interpretable IDS for healthcare IoT using SHAP-based feature explanations to ensure transparency in clinical environments.

Ullah *et al.* [28] introduced a transformer-based intrusion detection system (TNN-IDS) for MQTT-enabled IoT networks, using self-attention to learn patterns in MQTT traffic for attack detection. They evaluated the model on the MQTT-IoT-IDS2020 dataset and reported strong performance compared with multiple ML/DL baselines, highlighting the transformer’s ability to capture complex dependencies in MQTT communications. Zeghida *et al.* [29] proposed XMID-MQTT, an explainable ML–based intrusion detection framework for the MQTT protocol in IoT environments. Using the MQTT-IoT-IDS2020 dataset, the authors evaluated five classifiers (RF, linear SVM, RBF-SVM, CNN, and CNN-LSTM) to classify normal traffic and

multiple MQTT attack types (e.g., brute force and scanning variants), reporting that RF achieved the best performance (≈99.9% accuracy). To improve transparency and trust in IDS decisions, XMID-MQTT integrates explainable AI (XAI) techniques, specifically SHAP and LIME, to identify the most influential features behind model predictions and to provide interpretable insights for stakeholders. Allaga *et al.* [30] introduced MQTTEEB-D, a high-fidelity, real-world benchmark dataset for practical, real-time MQTT anomaly detection, collected from a physical IoT testbed rather than simulated traffic. The authors benchmarked a range of ML algorithms (including ensemble and boosting methods) under realistic constraints such as class imbalance, reporting performance differences between training on the original imbalanced data versus balanced variants. This work is valuable for MQTT/MQTT-SN IDS research because it emphasizes realistic data acquisition, reproducible benchmarking, and evaluation settings that better reflect operational IoT environments.

These recent efforts highlight the growing interest not only in improving detection accuracy but also in addressing practical deployment constraints such as latency, model explainability, and protocol-specific awareness—issues that our proposed framework also targets through its lightweight, deployable, and MQTT/MQTT-SN-specific design. Although several studies have explored intrusion/anomaly detection for IoT networks, MQTT/MQTT-SN traffic remains understudied, particularly in terms of protocol-aware feature engineering and standardized evaluation. Existing works often rely on general network features or narrow attack settings, which limit reproducibility and make it difficult to compare methods fairly. Therefore, there is a need for a framework that derives protocol-level features of MQTT, MQTT-SN, and evaluates multiple intelligent models under consistent experimental settings.

3. PROPOSED MODEL

The proposed framework aims to detect anomalies in MQTT and MQTT-SN protocol traffic using intelligent ML models. This section outlines the full development process. Figure 1 presents the end-to-end workflow of our system, progressing from top to bottom, from raw data to real-world operation. We begin with data collection and annotation, where traffic and events are captured and labeled to establish ground truth. Next, feature extraction and engineering transform raw records into discriminative inputs. Model design and algorithm implementation selects and implements candidate learners, followed by model training and validation, which tunes hyperparameters and verifies generalization using held-out and cross-validation splits.

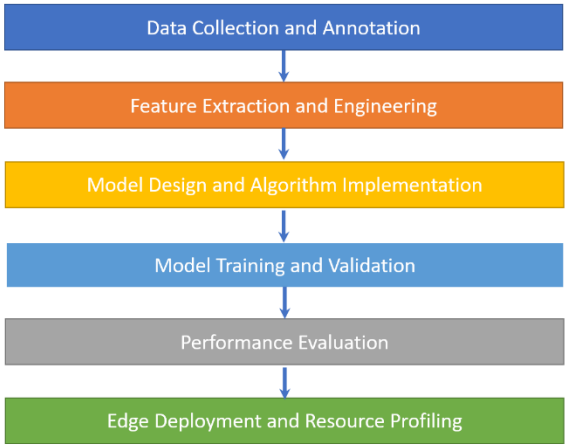

Figure 1. End-to-end pipeline for ML model development and deployment

In performance evaluation, models are compared using metrics such as accuracy, precision/recall, F1-score, receiver operating characteristic–area under the curve (ROC-AUC), and latency. Finally, edge deployment and resource profiling packages the selected model for constrained devices, measuring CPU, memory, energy, and inference time to ensure reliable, real-time operation. which consists of the following phases:

3.1. Data collection and annotation

To simulate realistic IoT communication, we designed a hybrid testbed comprising both MQTT and MQTT-SN brokers, several clients (sensors), and an attacker node. The following tools and devices were used:

– Broker: Eclipse Mosquitto for MQTT, IBM's RSMB for MQTT-SN,

- Clients: Python-based clients using the paho-mqtt library, and embedded ESP32 modules,
- Traffic generator: custom scripts generating normal MQTT/MQTT-SN payloads (e.g., temperature, motion, and light sensor readings) at regular intervals,
- Attack simulation: crafted using tools like hping3, Scapy, and custom malformed payloads.

To ensure reproducibility and clarify the dataset used in this study, we generated a labeled MQTT/MQTT-SN traffic dataset from a controlled hybrid IoT testbed. We captured all communications using Wireshark and tcpdump in PCAP format. The testbed included Eclipse Mosquitto as the MQTT broker and IBM RSMB as the MQTT-SN broker, with benign clients implemented as Python publishers/subscribers (paho-mqtt) and embedded ESP32 modules transmitting periodic telemetry (e.g., temperature, motion, and light readings). Malicious traffic was produced by an attacker node using hping3, Scapy, and custom scripts to emulate protocol-relevant threats, including DoS flooding (rapid CONNECT/PUBLISH), spoofing (forged client identifiers and topic-hierarchy manipulation), and injection/malformed messages (protocol-violating or incomplete packets), as well as replay behavior (duplicate PUBLISH patterns). The captured traffic was parsed using a custom Python extractor to derive MQTT/MQTT-SN metadata and export labeled CSV records. Overall, the dataset contains 10,000 labeled instances, comprising benign and malicious samples across the following classes: normal, DoS, spoofing, malformed/injection, and replay. Traffic was collected over the full capture sessions in a laboratory LAN/Wi-Fi environment, and labeling was performed using ground-truth knowledge of the executed attack scripts and corresponding capture time windows/logs. For reproducibility, we will make the dataset artifacts available upon publication (including the PCAP-to-CSV parsing scripts and feature-extraction code) to enable other researchers to replicate the experimental results.

We generated approximately 10,000 labeled records**,** categorized into the following traffic classes:

- Normal: valid publish/subscribe operations,
- DoS: flooding the broker with rapid CONNECT or PUBLISH messages,
- Spoofing: using fake client IDs or manipulating the topic hierarchy,
- Malformed: incomplete or protocol-violating packets,
- Replay: duplicate PUBLISH messages with altered timestamps.

Traffic was captured using Wireshark and tcpdump and stored in PCAP format. A custom Python parser was developed to extract MQTT-level metadata from packets and export it to CSV format.

3.2. Feature extraction and engineering

Rather than relying solely on basic flow-level features, we extracted protocol-specific attributes that capture the behavioral characteristics of MQTT/MQTT-SN communication. Table 1 presents the description of the selected features used for MQTT traffic analysis, detailing their types and purposes. The numerical features include *packet_length*, representing the total MQTT packet size in bytes; *topic_length*, indicating the length of the topic string; *publish_interval*, measuring the time in milliseconds between messages from the same client; and *client_id_entropy*, reflecting the Shannon entropy of the client ID to detect potential spoofing. Categorical features include *msg_type*, which specifies the MQTT message type (such as CONNECT, PUBLISH, or SUBSCRIBE), and *qos_level*, denoting the QoS level (0, 1, or 2). Binary features capture specific flags and indicators: retain_flag for the retain bit in PUBLISH messages, dup_flag for detecting duplicate messages, and session_flag for indicating whether a clean session was established during a CONNECT. All features work together to provide a complete representation of MQTT characteristics for subsequent analysis.

Table 1 summarizes the dataset used in this study, which consists of 10,000 labeled MQTT and MQTT-SN traffic instances captured in a controlled laboratory environment using Wireshark and tcpdump (PCAP). The dataset includes both benign and malicious records (2,000 benign and 8,000 malicious) covering five classes—normal, DoS, spoofing, malformed/injection, and replay—generated within a hybrid testbed comprising Mosquitto (MQTT), IBM RSMB (MQTT-SN), paho-mqtt clients, and ESP32 devices. The table also indicates the planned reproducibility support: the dataset artifacts will be made available upon publication, along with the PCAP-to-CSV parsing and feature-extraction scripts.

Table 1. Dataset summary

Field	Value
Total samples	10,000
Benign/Malicious	2,000/8,000
Classes	Normal, DoS, spoofing, malformed/injection, and replay
Protocols	MQTT and MQTT-SN
Duration	Full capture sessions in a controlled lab environment
Capture tools	Wireshark and tcpdump (PCAP)
Environment	Mosquitto (MQTT), IBM RSMB (MQTT-SN), paho-mqtt clients, and ESP32
Availability	Upon publication (PCAP-to-CSV scripts and feature-extraction code will be shared for reproducibility)

3.3. Preprocessing

Several preprocessing measures were taken in preparation for modeling. Initially, variables with categorical features, such as msg_type and qos_level, were encoded using one-hot encoding to convert them to binary values that ML algorithms could use. Afterward, continuous features were rescaled using Min-Max scaling to ensure that all numerical features fall within a specific range. Missing values (e.g., where a topic is not present) were replaced with default protocol-compliant values to maintain data integrity. Feature selection was then performed using recursive feature elimination (RFE) and the RF estimator to determine the most important features, retaining only those ranked by importance, and obtaining an efficient feature list optimized for modeling. Table 2 presents a concise description of the selected MQTT/MQTT-SN traffic features, including each feature's name, data type, and its role in characterizing packet structure, session behavior, and potential spoofing or anomalous activity.

- One-hot encoding was applied to msg_type and qos_level,
- Continuous features were normalized using Min-Max scaling,
- Missing values (e.g., when topic is absent) were imputed with protocol-compliant defaults,
- RFE with RF was used to refine the feature set based on importance scores.

Table 2. Description of the selected feature

Feature name	Type	Description
packet_length	Numerical	Total size of the MQTT packet (bytes)
msg_type	Categorical	CONNECT, PUBLISH, and SUBSCRIBE
qos_level	Categorical	QoS level (0,1,2)
topic_length	Numerical	Length of the topic string
retain_flag	Binary	Retain bit set in PUBLISH messages
dup_flag	Binary	Duplicate message indicator
publish_interval	Numerical	Time (ms) between messages from the same client
client_id_entropy	Numerical	Shannon entropy of client ID (to detect spoofing)
session_flag	Binary	Clean session flag during CONNECT

3.4. Model design and algorithm implementation

To handle the structured and partially temporal nature of MQTT data, we employed three different learning paradigms:

3.4.1. eXtreme gradient boosting classifier

XGBoost is a powerful gradient boosting algorithm renowned for its speed and accuracy, particularly when working with structured datasets. The model was implemented using the xgboost Python package, with key hyperparameters configured to optimize performance: n_estimators was set to 100, max_depth to 5, and learning_rate to 0.1 to balance training speed and predictive power. Additionally, a subsample ratio of 0.8 was used to introduce randomness and reduce overfitting, while logloss served as the evaluation metric for binary classification tasks. Due to its efficiency in handling both categorical and continuous features, XGBoost is well-suited for tabular data applications. Its lightweight and fast nature also makes it an excellent choice for near-real-time detection systems.

3.4.2. Random forest classifier

RF is an ensemble learning approach based on bagging, a strategy known for its robustness to high-dimensional, and noisy datasets. We implemented classification using sklearn.ensemble. RF classifier, and selected the key hyperparameters to improve the model performance. The number of trees (n_estimators=150) was set, as 150 was sufficient for computational efficiency and model complexity. Trees were allowed to grow fully (max_depth=None), which helped establish complexity sufficient to capture intricate data patterns. The Gini impurity criterion (criterion='gini') was used to split nodes, and class weights (class_weight='balanced') were adjusted to mitigate the effects of class imbalance. In addition to its predictive capacity, RF also serves as a baseline for feature selection, using its native feature importance ranking to identify the most influential variables in the dataset.

3.4.3. Long short-term memory neural network

To detect temporal anomalies (especially to identify repeated PUBLISH/SUBSCRIBE patterns or interval-based attacks), we used the LSTM model. The specific input structure we used was to create a sliding window of the last 10 consecutive packets for each client. By doing so, the model could analyze the sequential dependencies in the data. In terms of model architecture, we employed a 2-LSTM-layer model, with the first LSTM layer containing 128 units. We used return_sequences=true to pass the full sequence output to the

second layer. The first layer also included a dropout layer with a 30% dropout rate to regularize the model. The second LSTM layer contained only 64 units. And finally, we included an output layer with a sigmoid activation function, as we were performing a binary classification task. During the training stage, we set the loss function to binary_crossentropy and used the Adam optimizer due to its superior convergence properties compared to gradient descent. The model was trained for 25 epochs, with a batch size of 32. Additionally, a 20% allocation of the data was reserved for validation to monitor the model's generalization. To support LSTM learning from sequential data, we reshaped the Mac data into the required [samples, timesteps, features] structure to enable the LSTM to learn temporal patterns in the MQTT traffic.

3.5. Model training and validation

All models were trained using a 70:30 split of training and test data. To ensure robustness, we applied:

- 5-Fold cross-validation for XGBoost and RF
- Early stopping for LSTM based on validation loss
- SMOTE for oversampling the minority attack classes in the training set

All experiments were conducted on a machine equipped with 16GB RAM, an Intel i7 processor, and an RTX 3060 GPU (for LSTM).

3.5.1. Performance metrics

Model performance was evaluated using:

Accuracy: overall correct classifications

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: proportion of true attacks among detected attacks

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall: ability to capture actual attacks

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-score: harmonic mean of precision and recall

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Confusion matrix: visual insight into TP, FP, TN, and FN

As shown in Figure 2, the confusion matrix illustrates the relationship between actual and predicted classifications, highlighting true positives, true negatives, false positives, and false negatives.

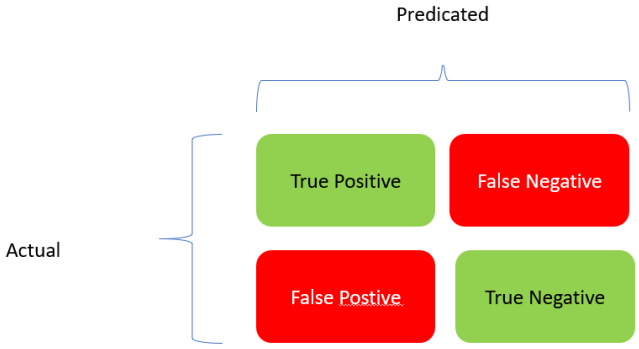

Figure 2. Confusion matrix

ROC-AUC curve: robustness against classification threshold

$$ROC - AUC = \int_0^1 TPR(FPR)d(FPR) \quad (5)$$

3.5.2. Deployment and resource profiling

To test real-world feasibility, the trained models were deployed on:

- ESP32 (Dual-core Xtensa, 4 MB Flash)
- Raspberry Pi Zero W (512 MB RAM and 1 GHz CPU)

The models were converted using:

- XGBoost to ONNX format
- LSTM to TensorFlow Lite

A lightweight Python agent listens to live MQTT traffic, extracts features, and performs model inference in real time.

Profiling metrics:

- Inference latency (ms)
- CPU and RAM usage during inference
- Average detection time per packet
- Power consumption (via USB measurement tool)

4. RESULTS AND DISCUSSION

This section presents the experimental results of applying three ML models—XGBoost, RF, and LSTM—on MQTT and MQTT-SN traffic for anomaly detection. The models were evaluated based on classification performance, computational efficiency, and real-world deployability on edge devices.

4.1. Classification performance

The performance of three models, XGBoost, RF, and LSTM, is summarized in Table 3, based on the MQTT/MQTT-SN test dataset and measured across five main policies. In conclusion, XGBoost achieved the best overall performance, scoring 97.4% accuracy, 96.1% precision, 95.8% recall, 95.9% F1-score, and 98.5% ROC-AUC, demonstrating strong and balanced detection performance. RF's performance was slightly lower at 96.8% accuracy, 94.5% precision, 94.2% recall, 94.3% F1-score, and 97.8% ROC-AUC, but it was still a competent performer overall. The LSTM model achieved the highest precision (97.0%) but the lowest recall (91.4%), resulting in an accuracy of 95.1%, an F1-score of 94.1%, and an ROC-AUC of 97.0%. Hence, the LSTM model is considered highly precise, with low sensitivity to relevant positive cases.

Table 3. Model performance on MQTT/MQTT-SN test dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
XGBoost	97.4	96.1	95.8	95.9	98.5
RF	96.8	94.5	94.2	94.3	97.8
LSTM	95.1	97.0	91.4	94.1	97.0

Observations:

- XGBoost achieved the highest overall performance, especially in accuracy and F1-score. It was particularly effective at handling both benign and malicious samples.
- RF performed comparably but was slightly less sensitive to minority-class detection (e.g., spoofing or malformed packets), likely due to limited interpretability at overlapping class boundaries.
- LSTM achieved high precision, detecting a high proportion of true attacks while generating fewer false alarms. However, its recall was slightly lower, likely due to limitations in sequence-based modeling for sparse anomalies.

4.2. Confusion matrix analysis

Table 4 shows the confusion matrix for the classification results, illustrating the model's effectiveness in distinguishing between normal and attack traffic. Out of all actual normal instances, 13,502 were correctly predicted as normal, while 311 were incorrectly classified as attacks (false positives). For actual attack instances, 10,321 were correctly identified as attacks, with 398 misclassified as normal (false negatives). These results demonstrate strong detection performance, characterized by high counts of true positives and true negatives. However, further reduction of false positives and false negatives would enhance the model's reliability in detecting MQTT/MQTT-SN attacks.

The model exhibits a false-positive rate of 2.2% and a false-negative rate of 3.7%, both of which are acceptable for real-time IoT monitoring. Misclassifications were most common for replay attacks that closely resemble legitimate traffic.

Table 4. Confusion matrix for XGBoost

	Predicted normal	Predicted attack
Actual normal	13,502	311
Actual attack	398	10,321

4.3. MQTT vs. MQTT-SN detection performance

Table 5 shows the protocol-level performance of the XGBoost model for MQTT and MQTT-SN traffic. For MQTT, the model achieved 98.1% accuracy and 96.5% F1-score, indicating highly reliable detection and balanced precision-recall performance. In comparison, the MQTT-SN protocol yielded slightly lower results, with an accuracy of 96.4% and an F1-score of 94.2%, indicating a strong overall performance but a marginal decrease in detection effectiveness compared to MQTT. These results highlight the model's robustness across both protocols, with particularly high effectiveness for standard MQTT traffic.

Table 5. Protocol-level performance (XGBoost)

Protocol	Accuracy (%)	F1-Score (%)
MQTT	98.1	96.5
MQTT-SN	96.4	94.2

Although both protocols achieved high scores, MQTT detection slightly outperformed MQTT-SN. This is attributed to MQTT's richer message metadata (headers and flags), which provide more useful features for classification. MQTT-SN's binary encoding and simpler structure made feature extraction less expressive.

4.4. Execution time and resource usage

Table 6 presents edge device model profiling results comparing the performance of XGBoost, RF, and LSTM (TFLite) on resource-constrained devices. XGBoost, running on a Raspberry Pi W, achieved the fastest inference time of 8.4 ms, with moderate CPU usage of 34% and RAM usage of 52 MB, making it the most efficient model in this evaluation. RF, also on a Raspberry Pi W, recorded a slightly longer inference time of 11.3 ms, higher CPU usage at 41%, and 57 MB of RAM usage. The LSTM model, deployed on an ESP32, exhibited the slowest inference time of 23.5 ms, with the highest CPU usage of 68% and RAM consumption of 78 MB, indicating greater computational demands compared to the other models. These results highlight XGBoost's suitability for real-time deployment on low-power edge devices.

Table 6. Edge device model profiling

Model	Device	Inference time (ms)	CPU usage (%)	RAM usage (MB)
XGBoost	Raspberry Pi W	8.4	34	52
RF	Raspberry Pi W	11.3	41	57
LSTM (TFLite)	ESP32	23.5	68	78

Observations:

- XGBoost offered the best trade-off between accuracy and latency on edge devices.
- RF incurred slightly more overhead without a performance gain.
- LSTM**,** although precise, had a higher computational burden, making it more suitable for gateway-level or cloud-hosted scenarios rather than ultra-low-power devices.

4.5. Discussion and implications

The results confirm that ML techniques can be successfully applied to detect anomalies in lightweight IoT communication protocols such as MQTT and MQTT-SN. Among all tested models:

- XGBoost is best suited for deployment in constrained environments due to its accuracy, efficiency, and scalability.
- LSTM is promising in contexts where temporal behavior is critical (e.g., detecting time-based flooding or periodic spoofing), but its cost must be weighed against system resources.
- The slight performance gap between MQTT and MQTT-SN highlights the need for further research into protocol-specific anomaly features**,** particularly for compact protocols like MQTT-SN.

Compared to previous IDS approaches for IoT, our framework achieves a balanced trade-off between detection performance and deployment feasibility. The use of MQTT/MQTT-SN-specific features enabled our

models—particularly XGBoost—to achieve over 97% accuracy while providing real-time inference on low-power devices. This contrasts with traditional ML-based IDS that rely on general network flows or require more powerful infrastructure. Moreover, by modeling temporal patterns using LSTM, we successfully detected anomalies such as replay and timing-based DoS attacks, which are often missed by static classifiers. These findings validate the added value of protocol-aware and lightweight anomaly detection in modern IoT security.

As shown in Table 7, our proposed framework achieves competitive accuracy compared to recent IoT IDS studies while also offering key advantages such as protocol-level feature modeling and lightweight deployment on real devices. Although some deep learning methods (e.g., GRUs and ensemble cascades) achieve slightly higher accuracy, they often require significant computational resources or cloud integration. Our framework strikes a balance between detection performance and deployability in constrained environments, addressing a critical gap in practical IoT security solutions.

Table 7. Benchmarking against state-of-the-art IDS approaches

Study and approach	Protocol	Algorithm/model	Accuracy (%)	Notes
Our work (XGBoost)	MQTT/MQTT-SN	XGBoost	97.4	Protocol-aware and real-time edge deployment
Francis *et al.* [14]	MQTT	Ensemble learning	99.1	No edge profiling; trained on SENMQTTSET
Palmese *et al.* [16]	MQTT/MQTT-SN	Adaptive QoS controller	N/A	Dynamic QoS adaptation for latency and hitrate optimization in IoT wireless sensor networks
Has *et al.* [15]	MQTT/MQTTS	Huffman, Zlib, LZW, Golomb–Rice, AES, TLS	N/A	Comparative analysis of compression and security techniques for efficient IoT data transmission
Hossain *et al.* [9]	General IoT security	Holistic security analysis and taxonomy	N/A	Comprehensive review of IoT vulnerabilities, attack taxonomy, communication security, service security, and forensic frameworks
Imran *et al.* [11]	MQTT	Correlation analysis+random forest/decision tree	99.81%	Correlation-based feature selection for efficient anomaly detection in MQTT IoT networks with reduced training and testing time for DoS, brute-force, and malformed attacks

4.6. Generalization to real-world internet of thing environments

While our experimental setup is based on a controlled testbed using synthetic traffic for MQTT and MQTT-SN protocols, we designed it to emulate realistic IoT communication patterns (e.g., periodic telemetry, burst publishing, and topic hierarchies). Nonetheless, we acknowledge the variability and complexity of real-world IoT systems, which often differ across application domains.

In smart homes, traffic typically involves intermittent device activity (e.g., motion sensors and door locks), making timing-based anomalies such as replay attacks detectable via our LSTM sequence modeling. In IIoT settings, high-frequency telemetry and process control messages may expose devices to DoS or spoofing, both of which are explicitly modeled in our detection pipeline. In healthcare IoT, where privacy and reliability are critical, our lightweight deployment on edge devices offers local security monitoring without relying on cloud inspection.

Furthermore, our use of protocol-aware features (e.g., QoS level, retain flag, topic length) is transport-layer agnostic, allowing the detection logic to be adapted to other MQTT-based deployments with minimal retraining. Given that MQTT is a standardized protocol across various sectors, we believe our model architecture and feature set offer strong generalization potential with minimal tuning.

Future work will involve testing our framework against real-world datasets, such as the TON_IoT, IoT-ID, or BoT-IoT repositories, and deploying it in live environments (e.g., campus or smart lab) to monitor actual IoT communications under both benign and adversarial conditions.

Furthermore, this study demonstrates the feasibility of on-device inference using optimized models and minimal processing power, paving the way for autonomous, edge-based security in IoT.

4.7. Ethical and legal considerations

The increasing deployment of AI-driven intrusion detection systems in IoT environments raises significant concerns related to privacy, data protection, and the responsible use of AI. In this study, all data was generated in a controlled laboratory testbed using synthetic device traffic that did not involve any real users, personal information, or privacy-sensitive content. This approach ensured full compliance with research ethics and avoided the risks associated with exposing identifiable data.

Nevertheless, for applied use, anomaly detection systems can operate on traffic data with metadata or associate patterns that are more indirectly related to user behavior, e.g., communication based on the timing of device usage or location-based communication. Our future deployment of the framework will consider the risks

and leverage data anonymization techniques, including, but not limited to, removing or hashing unique identifiers such as client IDs and IP addresses, and restricting access to training or logging data to authorized personnel only.

From a legal standpoint, deploying such systems must comply with applicable data protection laws or regulations governing the data collected, such as the GDPR in the EU or the country's privacy laws where deployment is planned. This includes obtaining informed consent, minimizing data collection, and ensuring that collected data does not exceed the intended security or research purpose.

Aligned with the beliefs of responsible AI, we will consider the importance of transparency surrounding model design when making decisions (for example, performing model evaluation with interpretable models using XGBoost and RF), efforts to limit bias in training datasets, and active ongoing monitoring of model performance and behavior in operational environments to ensure anomaly detection decisions are explainable, fair, and robust, particularly for sensitive sectors such as healthcare, or in IIoT (passive), into the commercial domain (active).

Moreover, as part of the responsible AI framework, we prioritize transparency in model decision-making (e.g., with interpretable models such as XGBoost and RF), bias mitigation in training datasets, and live monitoring and updating of model behavior. This embeds procedures into our anomaly detection decisions that are fair, robust, and understandable, especially in high-risk situations in healthcare and IIoT.

By applying these ethical safeguards throughout the design and deployment of our framework, we aim not only to provide a technically effective solution but also to deliver a socially responsible one.

5. CONCLUSION

This study presented a comprehensive ML-based framework for anomaly detection in MQTT and MQTT-SN traffic, aiming to enhance the security of lightweight IoT communication protocols. By extracting protocol-specific features and employing models such as XGBoost, RF, and LSTM, we demonstrated the feasibility of achieving high detection accuracy while maintaining efficiency suitable for edge deployment. Among the evaluated models, XGBoost achieved the best trade-off between accuracy, latency, and resource consumption, making it the most suitable choice for real-time intrusion detection on constrained devices.

Our experimental results further revealed that the detection performance was consistently high for both MQTT and MQTT-SN. However, slightly higher accuracy was achieved with MQTT due to the richer metadata availability. The confusion matrix analysis showed low false-positive and false-negative rates. At the same time, resource profiling confirmed the practicality of deploying the proposed models on devices such as the Raspberry Pi Zero W and ESP32. These findings validate the potential of intelligent learning models to provide lightweight, deployable, and effective security solutions for IoT ecosystems.

However, this study has certain limitations. First, while the dataset was comprehensive, it was generated in a controlled testbed environment, which may not fully capture the variability of real-world IoT deployments. Second, the feature set and models were optimized for MQTT and MQTT-SN so that additional tuning may be necessary for other IoT protocols or mixed-protocol environments. Lastly, although promising for temporal anomaly detection, the LSTM model was computationally demanding for ultra-low-power devices, limiting its direct deployment. Future work will focus on expanding the dataset to include real-world traffic, integrating adaptive learning mechanisms for evolving threats, and further exploring model compression techniques to improve deployment feasibility across diverse IoT hardware platforms. Future extensions of this work will focus on validating the detection framework using real-world IoT traffic datasets and live deployments across various domains, including smart home, healthcare, and industrial environments.

ACKNOWLEDGMENTS

The authors would like to acknowledge Princess Sumaya University for Technology (PSUT) and the Cybersecurity Research Centre at Universiti Sains Malaysia (USM) for their technical and administrative support in conducting this research.

FUNDING INFORMATION

This research received no external funding

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nabeel Mustafa Alassaf	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Selvakumar Manickam		✓				✓		✓	✓	✓	✓	✓		
Ammar Odeh	✓		✓	✓			✓			✓	✓		✓	✓
Mohammed Anbar	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable. This study did not involve humans or animals. All experiments were conducted using synthetic IoT network traffic generated in a controlled laboratory environment.

INFORMED CONSENT

Not applicable. This study did not involve humans or the collection of any personal or identifiable data.

DATA AVAILABILITY

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] J. M. Kizza, "Internet of Things (IoT): Growth, challenges, and security," *Guide to Computer Network Security*, 2024, pp. 557–573, doi: 10.1007/978-3-031-47549-8_25.
[2] J. S. Yalli, M. H. Hasan, and A. A. Badawi, "Internet of Things (IoT): Origins, embedded technologies, smart applications, and its growth in the last decade," *IEEE Access*, vol. 12, pp. 91357–91382, 2024, doi: 10.1109/ACCESS.2024.3418995.
[3] N. S. S, D. M. Anna, V. M N, and S. R. Kota, "Enabling lightweight device authentication in message queuing telemetry transport protocol," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 15792–15807, May 2024, doi: 10.1109/JIOT.2024.3349394.
[4] K. S. L. Kazi, "Transformation of agriculture effectuated by artificial intelligence-driven Internet of Things (AIIoT)," *Integrating Agriculture, Green Marketing Strategies, and Artificial Intelligence*, pp. 449–484, 2025.
[5] I. Imran, M. F. Zuhairi, S. M. Ali, Z. Shahid, M. M. Alam, and M. M. Su'ud, "Realtime feature engineering for anomaly detection in IoT based MQTT networks," *IEEE Access*, vol. 12, pp. 25700–25718, 2024, doi: 10.1109/ACCESS.2024.3363889.
[6] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 167–185, 2024, doi: 10.1016/j.iotcps.2023.12.003.
[7] A. Qaddos, M. U. Yaseen, A. S. Al-Shamayleh, M. Imran, A. Akhunzada, and S. Z. Alharthi, "A novel intrusion detection framework for optimizing IoT security," *Scientific Reports*, vol. 14, no. 1, Sep. 2024, doi: 10.1038/s41598-024-72049-z.
[8] M. Swain, N. Tripathi, and K. Sethi, "Identifying communication sequence anomalies to detect DoS attacks against MQTT," *Computers & Security*, vol. 157, Oct. 2025, doi: 10.1016/j.cose.2025.104526.
[9] M. Hossain, G. Kayas, R. Hasan, A. Skjellum, S. Noor, and S. M. R. Islam, "A holistic analysis of Internet of Things (IoT) security: Principles, practices, and new perspectives," *Future Internet*, vol. 16, no. 2, Jan. 2024, doi: 10.3390/fi16020040.
[10] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A review on the study on MQTT security challenge," in *2020 IEEE International Conference on Smart Cloud (SmartCloud)*, Nov. 2020, pp. 128–133, doi: 10.1109/SmartCloud49737.2020.00032.
[11] I. Imran, M. F. A. Zuhairi, S. M. Ali, Z. Shahid, M. M. Alam, and M. M. Su'ud, "Improving reliability for detecting anomalies in the MQTT network by applying correlation analysis for feature selection using machine learning techniques," *Applied Sciences*, vol. 13, no. 11, Jun. 2023, doi: 10.3390/app13116753.
[12] T. K. Boppana and P. Bagade, "GAN-AE: An unsupervised intrusion detection system for MQTT networks," *Engineering Applications of Artificial Intelligence*, vol. 119, Mar. 2023, doi: 10.1016/j.engappai.2022.105805.
[13] A. Diallo, L. Affognon, C. Diallo, and E. C. Ezin, "Towards the implementation of a dynamic IDS for IoT: Anomaly detection in MQTT traffic," in *Interdisciplinary Solutions for Underserved Areas (InterSol 2024)*, 2025, pp. 183–193, doi: 10.1007/978-3-031-86493-3_15.
[14] G. T. Francis, A. Souri, and N. İnanç, "A hybrid intrusion detection approach based on message queuing telemetry transport (MQTT) protocol in industrial internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 9, Sep. 2024, doi: 10.1002/ett.5030.
[15] M. Has, D. Kreković, M. Kušek, and I. P. Žarko, "Efficient data management in agricultural IoT: Compression, security, and MQTT

protocol analysis," *Sensors*, vol. 24, no. 11, May 2024, doi: 10.3390/s24113517.
[16] F. Palmese, A. E. C. Redondi, and M. Cesana, "Adaptive quality of service control for MQTT-SN," *Sensors*, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228852.
[17] J. Roldán-Gómez, J. Carrillo-Mondéjar, J. M. C. Gómez, and J. L. M. Martínez, "Security assessment of the MQTT-SN protocol for the Internet of Things," *Journal of Physics: Conference Series*, vol. 2224, no. 1, Apr. 2022, doi: 10.1088/1742-6596/2224/1/012079.
[18] R. Bin Mofidul, M. M. Alam, M. H. Rahman, and Y. M. Jang, "Real-time energy data acquisition, anomaly detection, and monitoring system: Implementation of a secured, robust, and integrated global IIoT infrastructure with edge and cloud AI," *Sensors*, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228980.
[19] S. Gupta, T. Sacchetti, and B. Crispo, "End-to-end encryption for securing communications in industry 4.0," in *2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, Dec. 2022, pp. 153–158, doi: 10.1109/MENACOMM57252.2022.9998272.
[20] K. Hwang, J. M. Lee, and I. H. Jung, "Performance monitoring of MQTT-based messaging server and system," *Journal of Logistics, Informatics and Service Science*, vol. 9, no. 1, pp. 85–96, Jan. 2022, doi: 10.33168/LISS.2022.0107.
[21] H. Siddharthan, T. Deepa, and P. Chandhar, "SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33095–33110, 2022, doi: 10.1109/ACCESS.2022.3161566.
[22] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, vol. 2019, no. 1, Jan. 2019, doi: 10.1155/2019/6516253.
[23] Z. Cai, Y. Si, J. Zhang, L. Zhu, P. Li, and Y. Feng, "Industrial Internet intrusion detection based on Res-CNN-SRU," *Electronics*, vol. 12, no. 15, Jul. 2023, doi: 10.3390/electronics12153267.
[24] J. Vansiya, A. Chandi, and R. A. Khan, "AI-based intrusion detection & prevention models for smart home IoT systems: A literature review," *Journal of Computer Science and Technology Studies*, vol. 7, no. 3, pp. 982–996, May 2025, doi: 10.32996/jcsts.2025.7.3.110.
[25] M. F. Almufareh, M. Humayun, Z. Ahmad, and A. Khan, "An intelligent LoRaWAN-based IoT device for monitoring and control solutions in smart farming through anomaly detection integrated with unsupervised machine learning," *IEEE Access*, vol. 12, pp. 119072–119086, 2024, doi: 10.1109/ACCESS.2024.3450587.
[26] A. Al Hanif and M. Ilyas, "Effective feature engineering framework for securing MQTT protocol in IoT environments," *Sensors*, vol. 24, no. 6, Mar. 2024, doi: 10.3390/s24061782.
[27] K. S. Alketbi and A. Mehmood, "A comprehensive survey of explainable artificial intelligence techniques for malicious insider threat detection," *IEEE Access*, vol. 13, pp. 121772–121798, 2025, doi: 10.1109/ACCESS.2025.3587114.
[28] S. Ullah *et al.*, "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks," *Computer Networks*, vol. 237, Dec. 2023, doi: 10.1016/j.comnet.2023.110072.
[29] H. Zeghida, M. Boulaiche, R. Chikh, A. Patel, A. L. B. Barros, and A. M. Bamhdi, "XMID-MQTT: explaining machine learning-based intrusion detection system for MQTT protocol in IoT environment," *International Journal of Information Security*, vol. 24, no. 3, Jun. 2025, doi: 10.1007/s10207-025-01036-w.
[30] H. Allaga, M. Biniz, and A. Farchane, "MQTTEEB-D: A high-fidelity benchmark for real-time MQTT anomaly detection using machine learning techniques," *Ad Hoc Networks*, vol. 181, Feb. 2026, doi: 10.1016/j.adhoc.2025.104062.

BIOGRAPHIES OF AUTHORS

Nabeel Mustafa Alassaf received his bachelor's degree in computer science from the University of Jordan in Jordan in 2002. He received his master's degree in computer science from the University of Jordan in Jordan in 2005. Currently, he is doing Ph.D. in network resource management at the Cybersecurity Research Centre, Universiti Sains Malaysia (USM). He is currently a lecturer with the computer science department at the University of Jordan with a master's degree. His research interest includes computer networks, internet communication protocols (IPv6), network security, mobile agents, cybersecurity, AI, and IoT. He can be contacted at email: nabeelalassaf@student.usm.my.

Prof. Dr. Selvakumar Manickam is the Director of the Cybersecurity Research Centre and a Professor specializing in cybersecurity, the Internet of Things, Industry 4.0, cloud computing, big data, and machine learning. He has authored and co-authored more than 220 articles in journals, conference proceedings, and book reviews. He has graduated 18 Ph.D. students in addition to master's and bachelor's students. He has given several keynote speeches and dozens of invited lectures and workshops at conferences, international universities, and industry. He can be contacted at email: selva@usm.my.

Ammar Odeh received his Ph.D. from the University of Bridgeport (UB), USA, in 2015. He is an Associate Professor at the Department of Computer Science, Faculty of King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan. His research interests include cybersecurity, cryptography, and the IoT. He can be contacted at email: a.odeh@psut.edu.jo.

Dr. Mohammed Anbar received the bachelor's degree in computer system engineering from Al-Azhar University, Palestine, the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), Malaysia, and the Ph.D. degree in advanced internet security and monitoring from the University Sains Malaysia (USM). His research interests include malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the internet of things (IoT), and IPv6 security. He can be contacted at email: anbar@cyres.usm.my.